Amendments to the Claims:

Rewrite the claims as set forth below. This listing of claims replaces all prior versions and listings of claims in the application:

1. (Previously Presented) A circuit for monitoring and resetting a co-processor comprising:

a hang detector module operative to detect a hang in the co-processor by detecting a discrepancy between a current state of the co-processor and a current activity of the co-processor; and

a selective processor reset module operative to selectively reset the co-processor without resetting a processor, in response to detecting a hang in the co-processor.

2. (Previously Presented) The circuit of claim 1 wherein an operating system executes on the processor.

3. (Cancelled)

4. (Previously Presented) The circuit of claim 1 wherein the discrepancy is detected by detecting the current state to be busy, by detecting a busy flag to be set, and detecting no progress on current activity, by detecting the same contents in a co-processor register as examined before and after a wait period.

5. (Previously Presented) The circuit of claim 1 further comprising:

a halt communications module operative to halt command communications with the co-processor, in response to detecting a hang in the co-processor;

a reset check module operative to detect if the co-processor has been successfully reset, in response to the resetting of the co-processor; and

a restart communications module operative to restart command communications with the co-processor, in response to detecting that the co-processor has been successfully reset.

6. (Previously Presented) A method of monitoring and resetting a co-processor comprising:

detecting a hang in the co-processor by detecting a discrepancy between a current state of the co-processor and a current activity of the co-processor; and

selectively resetting the co-processor without resetting a processor, in response to detecting a hang in the co-processor.

7. (Original) The method of claim 6 wherein an operating system executes on the processor.

8. (Cancelled)

9. (Previously Presented) The method of claim 6 wherein the discrepancy is detected by detecting the current state to be busy, by detecting a busy flag to be set, and detecting no progress on current activity, by detecting the same contents in a co-processor register as examined before and after a wait period.

10. (Previously Presented) The method of claim 6 further comprising:

halting command communications with the co-processor, in response to detecting a hang in the co-processor;

detecting if the co-processor has been successfully reset, in response to the resetting of the co-processor; and

restarting command communications with the co-processor, in response to detecting that the co-processor has been successfully reset.

11. (Currently Amended) A circuit for monitoring and resetting a co-processor comprising:

a hang detector module operative to detect a hang in the co-processor;

a halt communications module operative to halt ~~command~~ executable instruction communications with the co-processor, in response to detecting a hang in the co-processor;

a selective processor reset module operative to selectively reset the co-processor without resetting a processor, in response to detecting a hang in the co-processor;

a reset check module operative to detect if the co-processor has been successfully reset, in response to the resetting of the co-processor; and

a restart communications module operative to restart ~~command~~ executable instruction communications with the co-processor, in response to detecting that the co-processor has been successfully reset.

12. (Original) The circuit of claim 11 wherein the processor is a host processor and the co-processor is a graphics processor.

13. (Original) The circuit of claim 12 wherein the hang detector module detects the hang in the graphics processor by detecting a discrepancy between a current state of the graphics processor and a current activity of the graphics processor.

14. (Previously Presented) A system for monitoring and resetting a co-processor comprising:

a processor;

a co-processor; and

a memory containing instructions including:

hang detector module instructions operative to cause the processor to detect a hang in the co-processor by detecting the current state to be busy, as reflected in a busy flag, and detecting no progress on current activity, as reflected in the absence of co-processor register activity; and

selective processor reset module instructions operative to cause the processor to selectively reset the co-processor without resetting the processor, in response to detecting a hang in the co-processor.

15. (Previously Presented) The system of claim 14 wherein the processor is a host processor and the memory contains operating system instructions.

16. (Cancelled)

17. (Previously Presented) The system of claim 14 further comprising:

reset check module instructions operative to cause the processor to detect if the co-processor has been successfully reset, in response to the resetting of the co-processor.

18. (Previously Presented) The system of claim 14 further comprising:

halt communications module instructions operative to cause the processor to halt command communications with the co-processor, in response to detecting a hang in the co-processor; and

restart communications module instructions operative to cause the processor to restart command communications with the co-processor, in response to detecting that the co-processor has been successfully reset.

19. (Previously Presented) A system for monitoring and resetting a co-processor comprising:

a host processor;

a graphics processor; and

a memory containing instructions including:

hang detector module instructions operative to cause the processor to detect a hang in the graphics processor by detecting the current state of the graphics processor to be busy, as reflected in a graphics processor busy flag, and detecting the current activity of the graphics processor to be idle, as reflected in the absence of graphics processor register activity; and

selective processor reset module instructions operative to cause the processor to selectively reset the graphics processor without resetting the host processor, in response to detecting a hang in the graphics processor.

20. (Cancelled)

21. (Previously Presented) The system of claim 19 further comprising:

halt communications module instructions operative to cause the processor to halt rendering command communications with the graphics processor, in response to detecting a hang in the graphics processor;

reset check module instructions operative to cause the processor to detect if the graphics processor has been successfully reset, in response to the resetting of the graphics processor; and

restart communications module instructions operative to cause the processor to restart rendering command communications with the graphics processor, in response to detecting that the graphics processor has been successfully reset.

22. (Previously Presented) A system for monitoring and resetting a co-processor comprising:

a host processor;

a graphics processor; and

a memory containing instructions including:

hang detector module instructions operative to cause the processor to detect a hang in the graphics processor by detecting the current state to be busy, by detecting a busy flag to be set, and detecting no progress on current activity, by detecting the same contents in a co-processor register as examined before and after a wait period;

halt communications module instructions operative to cause the processor to halt rendering command communications with the graphics processor by setting a send flag to an off state and setting a receive flag to an off state, in response to detecting a hang in the graphics processor;

save snapshot module instructions operative to cause the processor to save a snapshot of the hardware and software status including any one or more of the following: graphics register data, graphics command queue data, chipset info, and AGP bus status, in response to the detecting a hang in the graphics processor;

selective processor reset module instructions operative to cause the processor to selectively reset the graphics processor without resetting the host processor, in response to detecting a hang in the graphics processor;

reset check module instructions operative to cause the processor to detect if the graphics processor has been successfully reset, in response to the resetting of the graphics processor;

display mode switch module instructions operative to cause the processor to perform a display mode switch, in response to the selectively resetting of the graphics processor;

functioning check module instructions operative to cause the processor to detect if the graphics processor is fully functioning, in response to the resetting of the graphics processor;

restart communications module instructions operative to cause the processor to restart rendering command communications with the graphics processor by setting the send flag to an on state, in response to detecting that the graphics processor has been successfully reset; and

software rendering module instructions operative to cause the processor to dynamically switch to software rendering mode, in response to detecting any one or more of the following: a detection that the graphics processor has not been successfully reset and a detection that the graphics processor is not fully functioning.

23. (Previously Presented) The system of claim 22 further comprising:

hang resolved prompt module instructions operative to cause the processor to display a prompt indicating that a hang was detected and resolved, in response to detecting that the graphics processor is fully functioning;

report send prompt module instructions operative to cause the processor to display a prompt requesting an input as to whether a user wants to have an error report sent to a remote location, in response to detecting a hang in the graphics processor;

report send module instructions operative to cause the processor to send an error report to a remote location including the hardware and software status, in response to receiving a user request to send an error report to a remote location; and

hang unresolved prompt module instructions operative to cause the processor to display a prompt indicating that a hang was detected and cannot be resolved without a reset of the host processor being performed, in response to detecting one or more of the following: the graphics processor was not successfully reset and the graphics processor is not fully functioning.

24. (Previously Presented) A memory containing instructions executable on a processor that causes the processor to:

detect a hang in a co-processor by detecting a discrepancy between a current state of the co-processor and a current activity of the co-processor; and

selectively reset the co-processor without resetting the processor, in response to detecting a hang in the co-processor.

25. (Original) The memory of claim 24 further including instructions that causes the processor to:

detect if the co-processor has been successfully reset, in response to the resetting of the co-processor.

26. (Previously Presented) A circuit for monitoring and resetting a co-processor comprising:

a hang detector module operative to detect a hang in the co-processor by detecting a discrepancy between a current state of the co-processor and data in one or more storage elements associated with the co-processor, wherein the data in the one or more storage elements represents a current activity of the co-processor; and

a selective processor reset module operative to selectively reset the co-processor without resetting a processor, in response to detecting a hang in the co-processor.

27. (Previously Presented) The circuit of claim 26 wherein:

the one or more storage elements includes a co-processor register; and

the discrepancy is detected by detecting the current state to be busy, by detecting a busy flag to be set, and detecting no progress on current activity, by detecting the same contents in a co-processor register as examined before and after a wait period.

28. (Previously Presented) The circuit of claim 26 wherein the hang detector module is operative to determine if the data in the one or more storage elements reflects processing of instructions by the co-processor.

29. (Previously Presented) The circuit of claim 26, wherein the current state of the co-processor is represented by data stored in the one or more storage elements associated with the co-processor.

30. (Previously Presented) A method of monitoring and resetting a co-processor comprising:

detecting a hang in the co-processor by detecting a discrepancy between a current state of the co-processor and data in one or more storage elements associated with the co-processor, wherein the data in the one or more storage elements represents a current activity of the co-processor; and

selectively resetting the co-processor without resetting a processor, in response to detecting a hang in the co-processor.

31. (Previously Presented) The method of claim 30 wherein:

the one or more storage elements includes a co-processor register; and

the discrepancy is detected by detecting the current state to be busy, by detecting a busy flag to be set, and detecting no progress on current activity, by detecting the same contents in a co-processor register as examined before and after a wait period.

32. (Previously Presented) The method of claim 30 further comprising determining if the data in the one or more storage elements reflects processing of instructions by the co-processor.

33. (Previously Presented) The method of claim 30, wherein the current state of the co-processor is represented by data stored in the one or more storage elements associated with the co-processor.

34. (Previously Presented) A memory containing instructions executable on a processor that causes the processor to:

detect a hang in a co-processor by detecting a discrepancy between a current state of the co-processor and data in one or more storage elements associated with the co-processor, wherein the data in the one or more storage elements represents a current activity of the co-processor; and

selectively reset the co-processor without resetting the processor, in response to detecting a hang in the co-processor.

35. (New) The circuit of claim 1 wherein the discrepancy is detected by comparing data representing a current state of the co-processor with data representing a current activity of the co-processor.